Card game final overall. Below are features that have been implemented into card\_game.py based on assignment 1 2 and 3.

Game mods P V P and P V E (the Ai is made to play the highest value card it can. What is highest value ex if player has dragon 1, 2 ,3 it should play star and wipe field / play the highest costing card in its hand)

Player and computer.

Cards and deck.

25 unique cards and interactions. With a deck of 40 cards.

19 spells (including resources) and 6 monsters.

Using Emojis ^^^

Players may only play cards when it is their turn.

Some of the new added cards.

"type": "thunder", "emojis": ["⚡️"], "costs": [5]}, # destroys creatures

"type": "shield", "emojis": ["⛨"], "costs": [1]}, # makes you invincible for one turn.

"type": "attack", "emojis": ["🗡️"], "costs": [0], "damage": [1]}, # makes you deal 1 damage to the enemy.

"type": "random", "emojis": ["🌌"], "costs": [3]}, # adds a random card to your hand, ignoring 'adds card' attribute for now.

type": "draw", "emojis": ["🎴", "🎴🎴" , "🎴🎴🎴",], "costs": [5, 6, 8,]}, # draws 2 , 3, 4 cards from the deck, ignoring 'cards' attribute for now.

"type": "life steel", "emojis": ["🧛🏻"], "costs": [3]},#takes 3 life points from other player give it to your self.

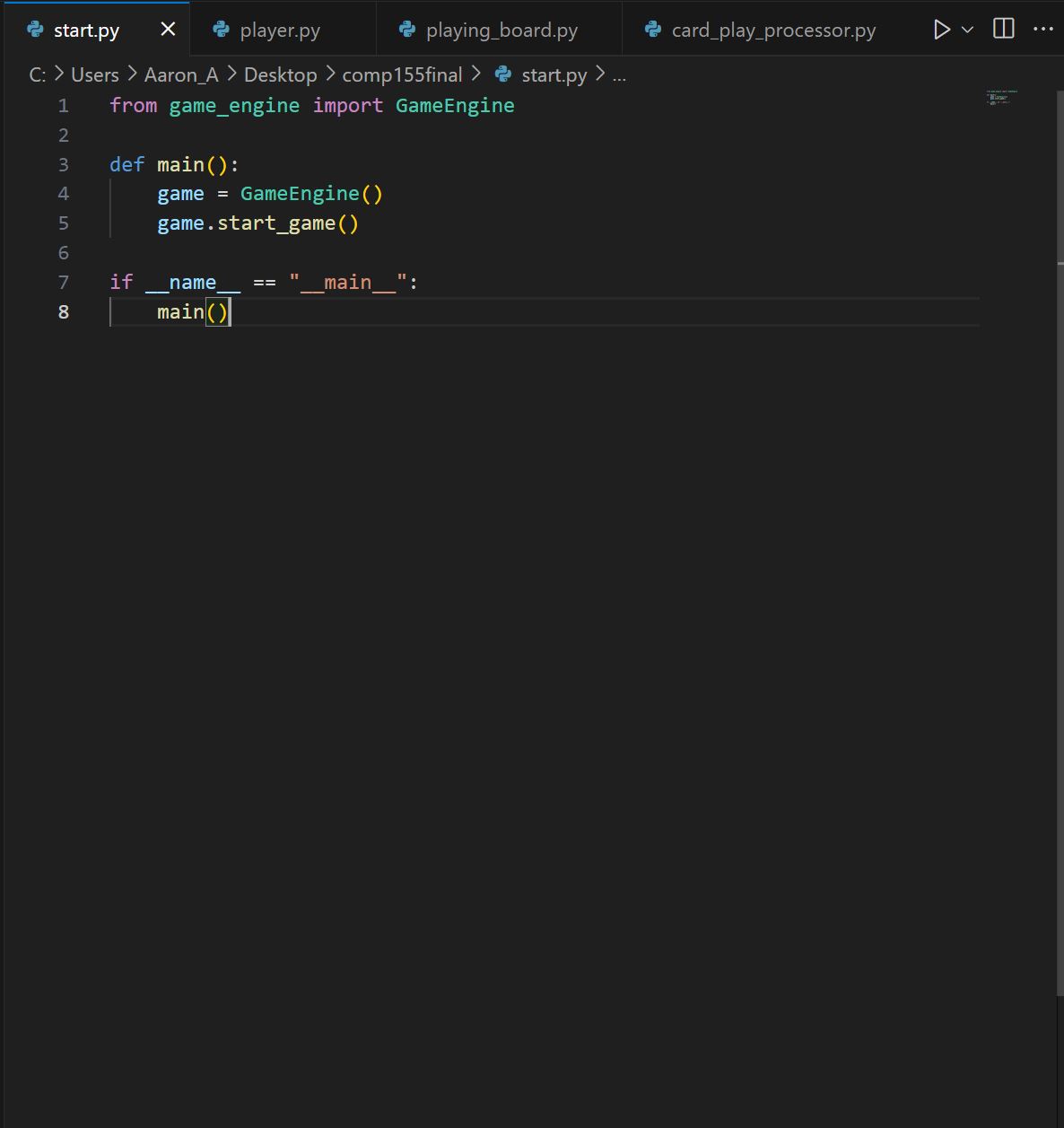
"type": "star", "emojis": ["☄️"], "costs": [3]},# clears the play field, like thunder but dose it for both fields

"type": "stop", "emojis": ["🚫"], "costs": [6]}, #prevents other player from playing any card for 1 turns.

"type": "dragon", "emojis": ["🐲", "🐲🐲", "🐲🐲🐲"], "costs": [1, 2, 3], "damage": [1, 2, 3]}, # deals damge when on fild = to damage can only be removed by star

And a Game engine. With 2-win conditions and logic tracking.

Overall, the full code is 494 lines long… this might not have been my biggest project, but each line of code contains more logic and meaning full impact to the output. Below you will find documentation for all the code. If I need to add more ill let you know at the end.



Not much here it just starts the game.

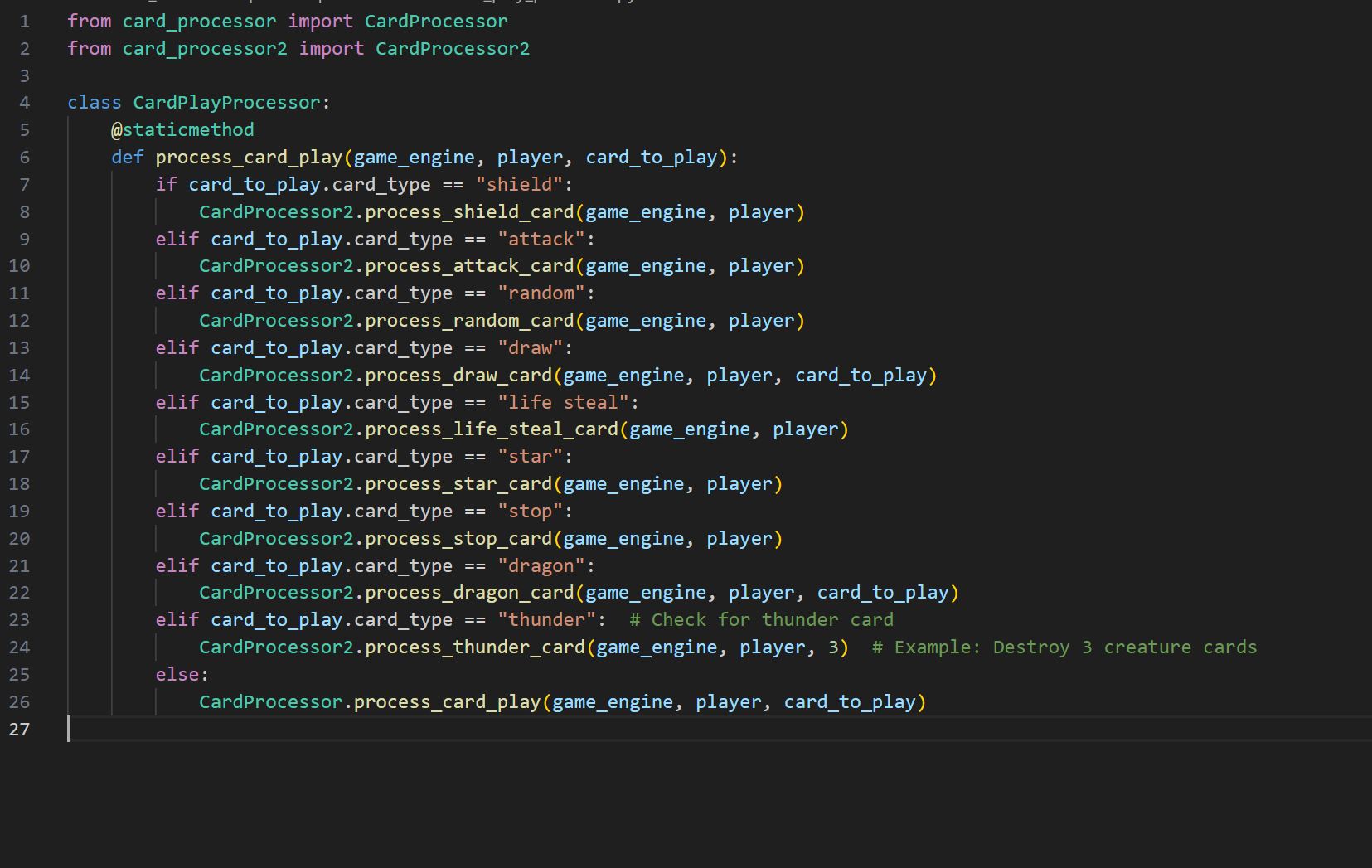
A computer screen shot of text

Description automatically generated

The Player class represents a player in the card game. It provides functionality for managing the player's hand, life points, and actions such as drawing cards and displaying the hand.

A screen shot of a computer program

Description automatically generatedThe PlayingBoard class is a game board where players can place their cards. It maintains separate fields for each player and provides methods to add cards to their respective fields and display the current state of the board.



The CardPlayProcessor class provides a static method to process card plays within the game engine.

The CardProcessor class provides static methods to process the play of different types of cards within a game. It handles various card types such as creature, resource, spell, and equipment, executing appropriate actions based on the type and level of the played card.

Methods

process\_card\_play(game\_engine, player, card\_to\_play)

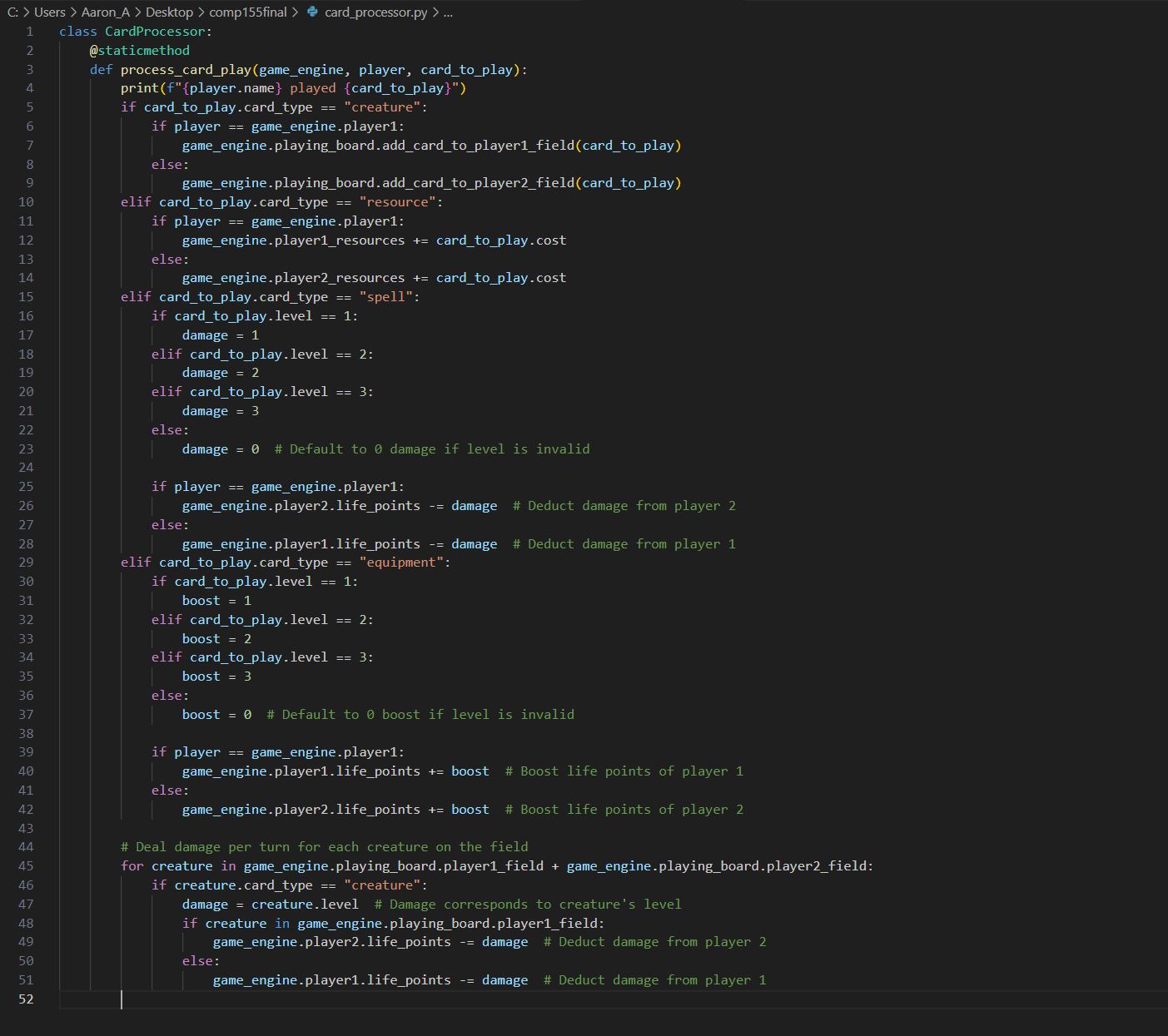
Description: Processes the play of a card within the game.

Parameters:

game\_engine (object): An instance of the game engine managing the game state.

player (object): The player who is playing the card.

card\_to\_play (object): The card object being played.



A screenshot of a computer program

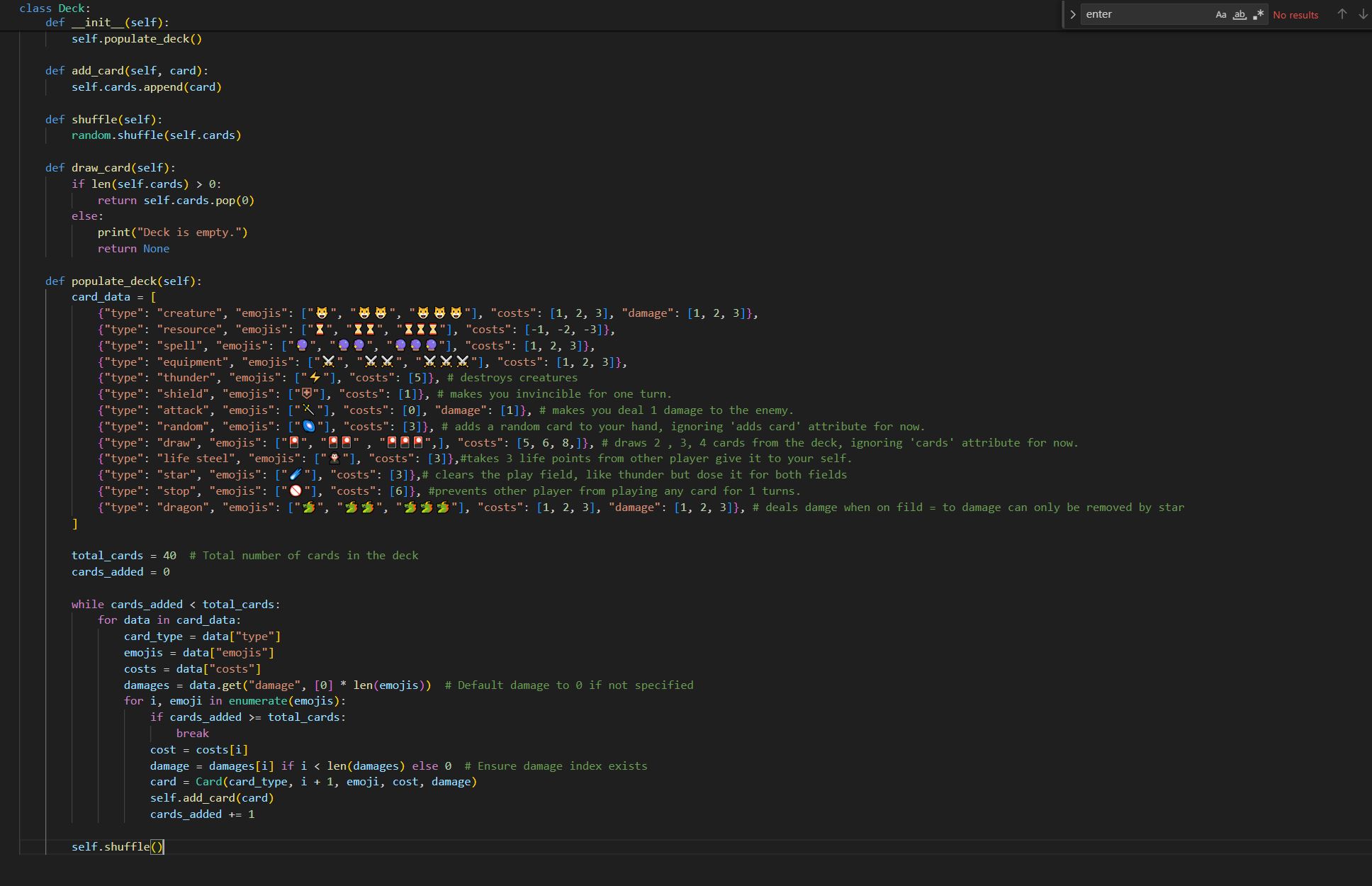
Description automatically generated



The CardProcessor2 class provides static methods to process the effects of various types of cards within a game. It handles different card types such as shield, attack, thunder, random, draw, life steal, star, stop, and dragon, executing appropriate actions based on the card type and game state.

Card and deck class represents a card in the card game. It has attributes such as card type, level, emoji representation, cost, and damage (if applicable). A screen shot of a computer code

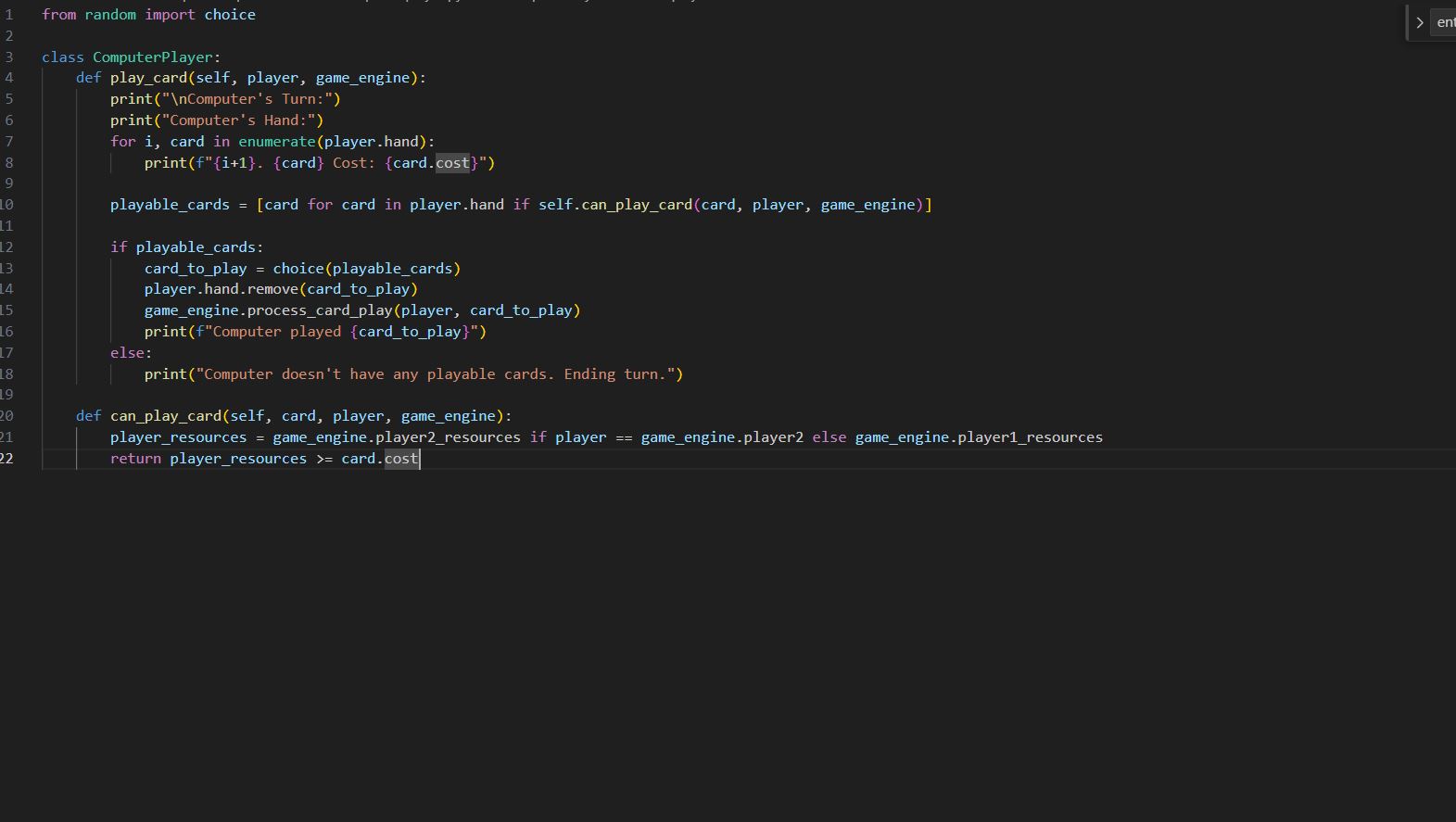
Description automatically generated

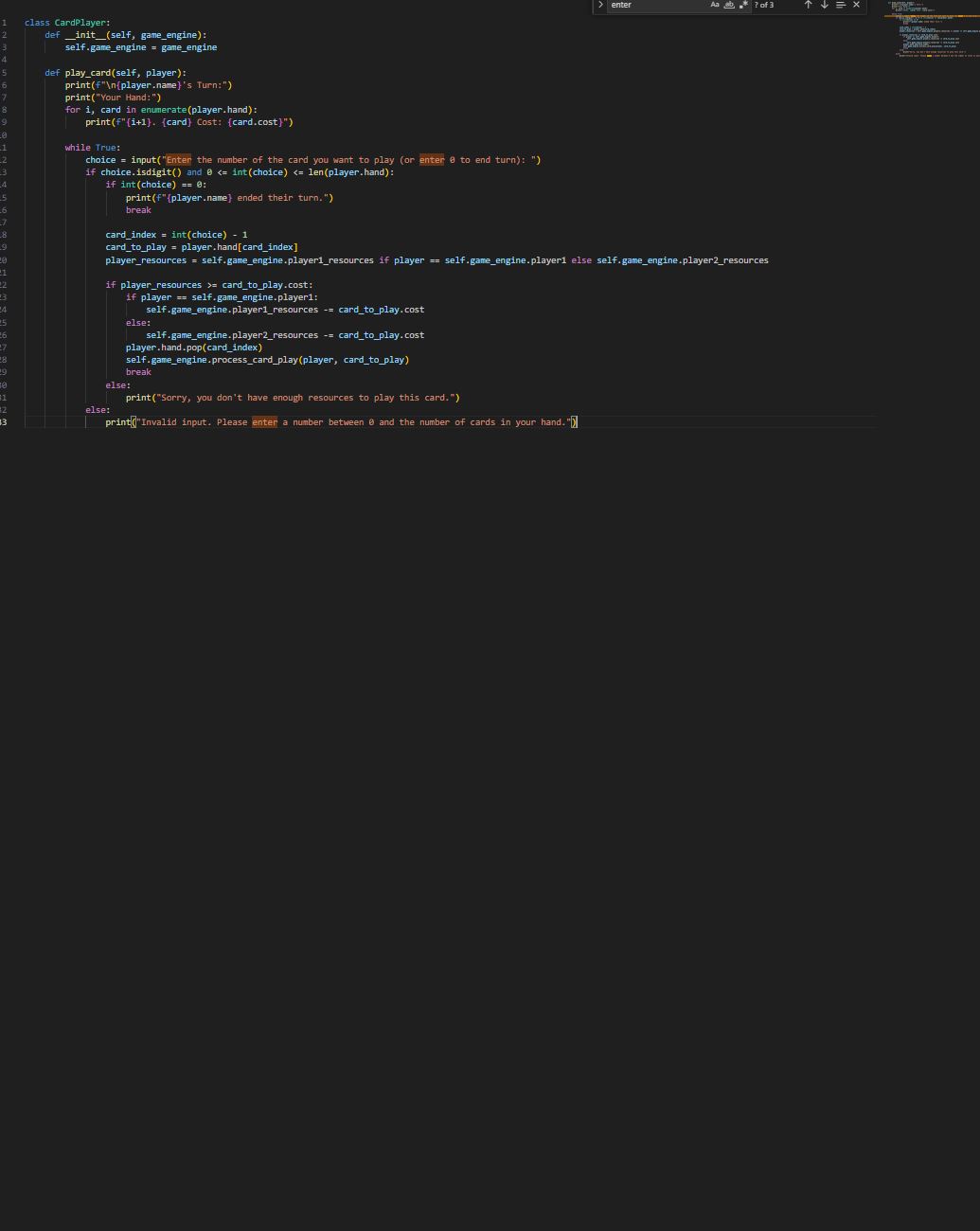


A computer screen shot of text

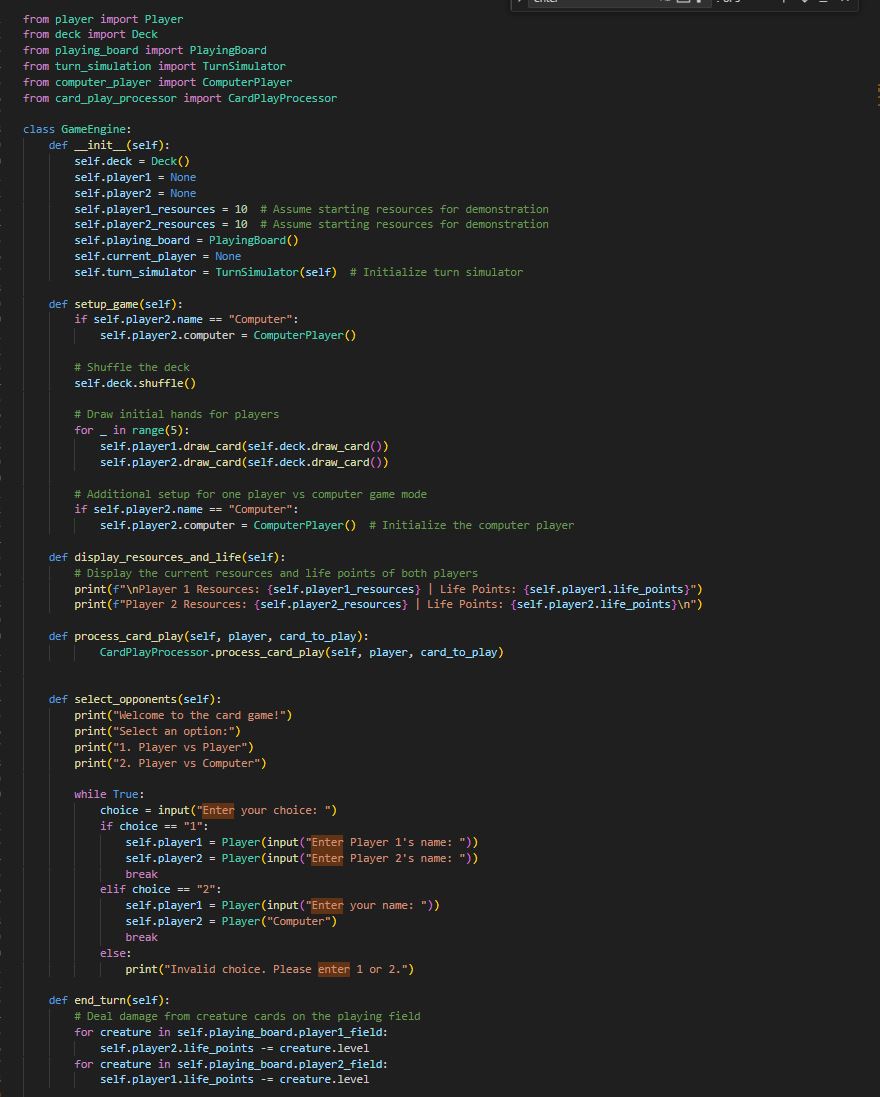
Description automatically generated

The TurnSimulator class simulates a turn within the card game. It handles the progression of a player's turn, including resource and life point updates, card drawing, and card playing.

The ComputerPlayer class represents a computer player in the card game. It provides a method to simulate the computer player's turn, including choosing a playable card from its hand and playing it.



The CardPlayer class represents a human player in the card game. It provides a method for the player to choose and play a card from their hand during their turn.



The GameEngine class keeps track of the game flow of the card game. It manages player setup, deck initialization, turn simulation, card processing, and game termination conditions.